APPENDIX A 1 of 4

From the Jul./Aug. 2004 issue of *IEEE Intelligent Systems*
PAGE: 1 | 2 | 3 | 4 | 5 | 6
Intelligent Systems

Authorization and Privacy for Semantic Web Services

Lalana Kagal and Tim Finin • University of Maryland
Massimo Paolucci, Naveen Srinivasan, and Katia Sycara • Carnegie Mellon University
Grit Denker • SRI International

Providing guarantees for security and privacy is paramount to the success of Semantic Web Services. In this article, the authors describe OWL-S policy annotations and extend the OWL-S Matchmaker and OWL-S Virtual Machine to support the processing of those policies.

Web Services will soon handle users' private information. They'll need to provide privacy guarantees to prevent this delicate information from ending up in the wrong hands. More generally, Web Services will need to reason about their users' policies that specify who can access private information and under what conditions.

These requirements are even more stringent for Semantic Web Services that exploit the Semantic Web to automate their discovery and interaction because they must autonomously decide what information to exchange and how.

In our previous work, we proposed ontologies for modeling the high-level security requirements and capabilities of Web Services and clients.[1] This modeling helps to match a client's request with appropriate services—those based on security criteria as well as functional descriptions. For example, a Web Service could state that it can perform OpenPGP encryption and requires an invoker that can authenticate itself and communicate in XML. We added functionality to the DAML-S Matchmaker[2] (an earlier version of the OWL-S Matchmaker) that lets it verify if a service's capabilities fulfill the invoker's

security requirements and vice versa. Our results assist coarse-grain matching decisions such as "Does the service provide encryption?" or "What kind of credential do I have to provide to authenticate myself to the service?"

In this article, we propose a more fine-grain security markup of service parameters in OWL-S. We extend our previous work with annotations about the security and privacy policies of services. We express these annotations in Rei, a logic-based language that lets you define rules and constraints over domain-specific ontologies.[3] Our work aims to provide security and policy annotations for OWL-S service descriptions and enforcements by extending the OWL-S Matchmaker for policy matching and the OWL-S Virtual Machine (VM)[4] with policy enforcement and security mechanisms.

next >>

Role of policies

Policies specify who can use a service and under which conditions, how information should be provided to the service, and how the provided information will be used. Policies should be part of Web Service representations—particularly those on the Semantic Web (see the "Related Work" sidebar for more background information).

In our work, a client-server model involves a client that wants to invoke a Web Service. We view the use of policies as *symmetric*—policies that constrain both the provider and requester. You can easily extend this model to a service-service architectural model.

Here, we address two kinds of policies: *privacy* and *authorization*. Privacy policies specify under what conditions you can exchange information and the legitimate uses of that information. For example, a privacy policy might say that a provider could give a requester a key to access private information only if the key is encrypted during transmission. When a requester discovers the policy, it should decide whether it can satisfy this condition. The requester might have its own privacy policy that requires keeping certain information confidential, so it likewise can't share unencrypted private information. The requestor's privacy policy prevents it from interacting with Web Services that don't perform the needed encryption.

Privacy policies help specify data confidentiality during transmission as well as after receipt. Consider a service that says it won't distribute details it receives as input. A requester that values privacy might see this as an important requirement.

You can interpret a Web Service's privacy policies as an obligation and contract. For example, if after invocation, a service does provide a requester's details to a telemarketer, the person represented by the requester could take legal action against the service on the basis of the policy. As financial transactions become more common among Web Services and as Web Services start dealing with confidential information (such as names, addresses, social security numbers (SSNs), credit cards, and telephone numbers), more people will expect the enforcement of privacy policies.

Authorization policies constrain the pro-vider to accept requests for service only from certain clients. For example, a service's authorization policy could state that a requester must act on behalf of a person who belongs to a certain organizational group and can prove membership with a digital certificate. Similarly, the requester could limit invocation to selected providers.

A motivating example

Consider a scenario in which a scientist is looking for an online computing service for her experimental data. Her privacy policy requires that any personal information provided to the service (such as name or SSN) stay confidential. So, she's only looking for Web Services that accept encrypted data and that don't release personal information to other services or agents.

The scientist finds a Web Service that can perform the necessary data computations. The service's authorization policy says that it allows access only to members of certain, selected organizations and that the scientist's registration must be authenticated.

In this article, we'll approach the formalization and processing of these privacy and authentication policies on two abstraction levels. On a more abstract level, we provide ontologies to annotate Web Service input and output parameters with security characteristics that state whether these parameters are encrypted or digitally signed, and we rely on Rei to formalize the privacy and authorization policies.

On a more concrete level, selecting Web Services that satisfy the requester's policies will be part of an extension of the OWL-S matchmaking process. Furthermore, cryptographic mechanisms such as encrypting or signing messages are enforced via integration into the OWL-S VM, a generic processor for the OWL-S process model and tool for automatic invocation of OWL services.

next »